UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

PLATAFORMA DE EVALUACIÓN DE ALGORITMOS DE DETECCIÓN DE PERSONAS.

Autor: Anthony Bryan Santiago Mendieta
Tutor: Álvaro García Martín
Ponente: José M. Martínez Sánchez

-TRABAJO FIN DE GRADO-

Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2017

PLATAFORMA DE EVALUACIÓN DE ALGORITMOS DE DETECCIÓN DE PERSONAS.

Autor: Anthony Bryan Santiago Mendieta
Tutor: Álvaro García Martín
Ponente: José M. Martínez Sánchez

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2017

Trabajo parcialmente financiado por el gobierno español bajo el proyecto TEC2014-53176-R (HA-Video)

Resumen.

El objetivo del proyecto ha sido elaborar una plataforma web automática de evaluación de algoritmos de detección de personas. Los algoritmos de detección se encuentran actualmente en auge y en este caso se ha elegido la temática de detección de personas. En primer lugar, se lleva a cabo un estudio de las plataformas ChangeDetection.net (CDNET) y People Detection Beanchmark repository (PDbm), del grupo de investigación Video Processing Understanding Lab (VPULab). Ambas plataformas estan orientadas a la evaluación de algoritmos. Tras el estudio de ambas plataformas se automatiza PDbm basándose en CDNET. El diseño del sistema se ha realizado en cuatro bloques funcionales: Servidor web, base de datos, aplicación de evaluación y gestor de correo. Además, se adopta el modelo de programación en capas en el servidor web.

Durante el desarrollo del proyecto se han implementado tecnologías específicas para cada bloque: Windows 7, Apache, MySQL, PHP, Matlab, Outlook. Posteriormente, se implementa un módulo de control para el administrador de la plataforma. Después, se prueba la plataforma en un enterno de pre-producción en local y se verifica que la funciona correctamente. Finalmente, se analiza la posibilidad de continuar desarrollando la plataforma de forma modular debido al diseño del sistema realizado.

En conclusión, se ha trabajado con aplicaciones de distintos tipos y dedicadas a distintos fines. Se ha logrado unirlas con el objetivo que formen un sistema único.

Palabras clave

Detección de personas, plataforma web, base de datos, evaluación de algoritmos.

Abstract.

The objective of the project has been to develop an automatic web platform for the evaluation of algorithms for detecting people. The detection algorithms are currently booming, in this case the topic of people detection has been chosen. First, it has been studied of the platforms ChangeDetection.net (CDNET) and People Detection Beanchmark repository (PDbm) of the Video Processing Understanding Lab (VPULab). Both platforms are oriented to the evaluation of algorithms. After studying both platforms, it was decided to automate PDbm based on CDNET. The design of the system has been made in four functional blocks: Web server, database, evaluation application and mail manager. In addition, we adopt the layered programming model on the web server.

During the development of the project, several specific technologies have been implemented for each block: Windows 7, Apache, MySQL, PHP, Matlab, Outlook. Subsequently, a control module is implemented for the platform administrator. Then, platform is tested in a local pre-production space and verified to be working properly. Finally, we analyze the possibility of continuing to develop the platform in a modular way due to the design of the system.

In conclusion, we have worked with applications of different types and dedicated to different purposes. It has been possible to unite them in order to form a unique system.

Keywords

People detection, web platform, databases, algorithm evaluation.

Agradecimientos.

Gracias a todos, en especial a mi hermano Alexander.
Anthony Bryan Santiago Mendieta.
Junio 2017.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

Los sistemas de video digitales han ido influyendo en la vida cotidiana durante los últimos años. Cada vez es más común utilizar un sistema de vídeo en múltiples ámbitos como la seguridad, ocio, etc. Paralelamente a ello, tambien se usan estos sistemas en el ámbito de la investigacion. A su vez, la falicidad de adquisición de sistemas de video digitales también ha incrementado de manera muy notable. Debido a esto, son más las personas dedicadas al estudio y análisis automático de secuencias de vídeo . Dentro de todos las etapas de análisis típicas de vídeo, una de ellas es la deteccion de personas. En este campo se intenta poder interpretar las imágenes de manera automática con el objetivo de poder sustituir la supervision humana. Por ello, debido a su gran influencia y utilidad que tiene en la sociedad, investigadores de diversas partes del mundo tratan de optimizar sus algoritmos. Existen actualmente infinidad de algoritmos, sistemas y propuestas para mejorar la detección automática de personas en secuencias de vídeo, y esto da lugar una gran cantidad de resultados.

Cada nuevo algoritmo del estado del arte debe ser evaluado sobre un dataset o conjunto de videos y una métrica de evaluación, de esta manera se consigue medir su efectividad. El problema reside en que cada autor o conjunto de autores suelen definir su propio protocolo de evaluación (dataset y métrica), por lo que es díficil comparar la eficiencia de dos o más algoritmos que han sido testeados con datasets distitnos.

La motivación de este trabajo de fin de grado será poder comparar distintos algoritmos de detección de personas bajo las mismas condiciones y poder establecer un ranking en función de distintas medidas de forma automática.

1.2. Objetivos

El objetivo principal de este trabajo es diseñar e implementar una plataforma de evaluación y comparación automátcia de algoritmos de detección de personas. Para ello y siguiendo la metodología propuesta para la evaluación y comparación automática de algoritmos de segmentación frente-fondo propuesta en ChangeDetection.net (CDNET): A video database for testing change detection algorithms[1]. [1]

En una primera etapa, se definirá el conjunto de secuencias, anotaciones o ground truth y métrica de evaluación. En segundo lugar, se establecerán los formatos que deben seguir los ficheros de entrada a nuestro sistema por parte de los usuarios. En tercer lugar,se procesarán los archivos recibidos para evaluarlos. Por último,se mostrarán los resultados de cada algoritmo en la interfaz web para que puedan ser visualizados por los usuarios de la página. Durante todo el proceso el administrador será informado del procesado de un nuevo algoritmos asi como de los resultados obtenidos. Una vez finalizado el proceso el administrador podrá disponer de todos los algoritmos y resultados procesados hasta el momento. Además, podrá seleccionar cuales de ellos serán visibles finalmente en la plataforma web.

Uno de los objetivos adicionales de la plataforma es que los usuarios puedan interactuar fácilmente con la interfaz web para que no tengan problemas en enviar sus datos. A su vez, el administrador tampoco deberá tener dificultades para mantener la plataforma.

La plataforma realiza la tarea de recogida de datos enviados por el usuario para su posterior almacenamiento, procesamiento y visualización. Además, integra la opción de manipular la visualizacion de la web de manera sencilla con un módulo de control destinado a un administrador.

Paralelamente a la implementacion de la plataforma, se tendrá presente durante todo el desarrollo que tanto el usuario como el administrador tengan que realizar la minima cantidad de acciones, a menos que requieran exclusivamente de su actividad. Con el objetivo de hacer lo más robusto posible el sistema o plataforma de evaluación de algoritmos de detección de personas, se hará un análisis de las distintas decisiones que se tomarán en función de la aparición de posibles errores en el sistema o por parte de los usuarios. Se estudiará como presentar los resultados a los visitantes de la plataforma para que puedan obtener una conclusión rápida del estado de los algoritmos evaluados.

[1] http://www.changedetection.net/

1.3. Estructura de la memoria

La memoria del proyecto se divide en los siguientes capítulos:

- Capítulo 1. Introducción: Motivación y objetivos del proyecto.
- Capítulo 2. Estado del arte: Plataformas de comparacion de algoritmos existentes, ChangeDetection.net (CDNET) y People Detection Benchmark repository (PDbm).
- Capítulo 3. Diseño: Sistema automático de evaluación. Diseño en bloques.
- Capítulo 4. Desarrollo plataforma evaluación: Servidor web (entorno WAMP), Matlab y Outlook.
- Capítulo 5. Resultados: Interaccion usuario e interaccion administrador. Resultado interfaz web.
- Capítulo 6. Conclusiones y trabajo futuro.
- Referencias y anexos.

Capítulo 2

Estado del arte

En este capitulo se describira en detalle la plataforma ChangeDetection.net (CDNET): A video database for testint change detecion algorithms. Se pondrá especial énfasis en la plataforma CDNET debido a que el objetivo del proyecto es replicar su sistema general y orientarlo a la evaluación de algoritmos de detección de personas. Por último, se detallará la plataforma PDbm[2] del grupo de investigación Video Processing and Understanding Lab (VPULab) de la universidad Autónoma de Madrid, donde se realizan evaluaciones de algoritmos de deteccion de personas de forma estática. Sobre esta plataforma se desarrollará el proyecto con la intención de transformarla y que realize los procesos de forma automática basándose en la plataforma CDNET.

2.1. Introducción

En la actualidad, el desarrollo de algoritmos de deteccion o reconocimiento crece exponencialmente. Gracias al notable desarrollo de internet en el mundo, la facilidad de compartir información ha incrementado en las últimas decadas. Son muchas las personas que intercambian frecuentemente opiniones, dudas y conocimientos en foros o portales web. Por consiguiente, debido a esta interaccion entre autores, el número de técnicas o algoritmos del estado del arte aumenta progresivamente. Paralelamente tambien se encuentra la motivación por la investigacion y el desarrollo impulsada en las universidades o centros de estudios superiores.

Como consecuencia, resulta atractivo poner en común los trabajos realizados por los distintos autores. De esta manera se favorece la competencia y la motivacion por la mejora de los algoritmos de cada autor.

Para lograr esta tarea se disponen de plataformas web dedicadas a la evaluacion

de algoritmos. Debido al aumento de número de autores y a la continua interactividad de la plataforma, estos sistemas tienden a ser automáticos.

Los plataformas actuales evaluan algoritmos de distintos tipos como: segmentacion frente-fondo, identificacion de usuario, etc. En este trabajo se evaluaran algoritmos de deteccion de personas.

La detección de personas es uno de los algoritmos que se desarrollan actualmente en muchas partes del mundo. Su utilidad reside en aplicaciones de videovigilancia, seguimiento de personas, etc.

2.2. ChangeDetection.net (CDNET)

En esta sección describieremos la plataforma web de evaluación ChangeDetection.net en la que nos basaremos a modo de ejemplo para automatizar la plataforma PDbm.

La plataforma web ChangeDetection se basa en la necesidad de segmentación de video como un primer filtro antes de su procesamiento. El procesamiento de video incluye aplicaciones como videovigilancia (conteo de personas, reconocimiento de acciones, detección de anomalías, etc.), entornos inteligentes (detección de caidas, detección ocupación aparcamientos, etc.), etc. En general, aplicaciones de vision artificial.

La funcionamiento de la plataforma sigue la figura 2.1:

Figura 2.1: Sistema CDNET

- La plataforma dispone de una serie de datasets que se ofrecen a los usuarios.
- El usuario se descarga los datasets y los evalúa con su algoritmo de segmentacion
- El usuario envía los resultados a la plataforma en un formato específico.
- La plataforma almacena los resultados enviados y los evalúa para ser publicados posteriormente. Estos procesos son automáticos.

.La plataforma CDNET se divide en 7 módulos diferentes en forma de pestañas donde el usuario puede navegar libremente según la información que desea saber. En la figura 2.2 se muestran los módulos.

Figura 2.2: Menú CDNET

En el primer módulo "HOME" se da una visión general de la plataforma. Empieza detallando los motivos que dieron lugar a la creación de esta plataforma. Después, se muestra una breve explicación de los datasets ofrecidos. Posteriormente, se describe concisamente cada uno de los 6 módulos restantes. Finalmente se menciona a las personas que organizan y mantienen esta plataforma, así como los reconocimientos a los que han colaborado en ella desde sus principios brindando su esfuerzo y apoyo. A continuación describiremos los módulos más importantes de la plataforma.

2.2.1. CDNET: DATASETS

Este módulo se divide en 3 submódulos, "OVERVIEW", "2012 DATASET" y "2014 DATASET". Se considera "Dataset" a un conjunto de videos. En el submódulo "OVERVIEW" se describen los datasets de manera detallada y se categorizan los videos en función del contenido del mismo. En la tabla 2.1 se muestran las distintas categorias.

Posteriormente, se define el "Ground truth", regiones de los frames que han sido etiquetadas manualmente por personas como zona de interés. En este caso, se etiqueta el frente y el fondo de un frame. En la figura 2.3 se puede apreciar el ejemplo que muestra CDNET.

El submódulo "2012 DATASET" y "2014 DATASET" muestran el mismo tipo de información. Primeramente, se detallan los aspectos técnicos del Dataset. Después, se muestran enlaces para descargar el dataset completo, categorias completas o videos individuales. Los videos individuales aparecen agrupados por la clasificacion mencionada en la tabla 2.1. Para cada video, si se coloca el cursor sobre el frame de ejemplo, se muestra un frame aleatorio del video, el "Ground Truth" de dicho frame y la cantidad de frames del video. Todos los videos o el dataset completo se puede descargar en formato zip o 7z. El submódulo "2014 DATASET" es una extesión de "2012 DATASET", se añaden 22 videos y 5 nuevas categorías. En la figura 2.4 se muestran los videos de la categoria "Baseline", así como las demas categorias y los

Categories	Description
Baseline	Category represents a mixture of mild challenges typical of the next 4 categories. Some videos have subtle background motion, others have isolated shadows, some have an abandoned object and others have pedestrians that stop for a short while and then move away. These videos are fairly easy, but not trivial, to process, and are provided mainly as reference.
Dynamic Background	Category includes scenes with strong (parasitic) background motion: boats on shimmering water, cars passing next to a fountain, or pedestrians, cars and trucks passing in front of a tree shaken by the wind.
Camera Jitter	Category contains indoor and outdoor videos captured by unstable (e.g., vibrating) cameras. The jitter magnitude varies from one video to another.
Intermittent Object Motion	Category includes videos with scenarios known for causing "ghosting" artifacts in the detected motion, i.e., objects move, then stop for a short while, after which they start moving again. Some videos include still objects that suddenly start moving, e.g., a parked vehicle driving away, and also abandoned objects. This category is intended for testing how various algorithms adapt to background changes.
Shadows	Category consists of indoor and outdoor videos exhibiting strong as well as faint shadows. Some shadows are fairly narrow while others occupy most of the scene. Also, some shadows are cast by moving objects while others are cast by trees and buildings.
Thermal	Category includes videos that have been captured by far-infrared cameras. These videos contain typical thermal artifacts such as heat stamps (e.g., bright spots left on a seat after a person gets up and leaves), heat reflection on floors and windows, and camouflage effects, when a moving object has the same temperature as the surrounding regions.
Challenging Weather	Category includes outdoor videos captured in challenging winter weather conditions, i.e., snow storm, snow on the ground, fog.
Low Frame-Rate	Category contains videos capture at varying frame-rates between 0.17 fps and 1 fps.
Night	Category includes videos captured at night (difficult light conditions) of, primarily, motor traffic.
PTZ	Category contains video footage captured by pan-tilt-zoom cameras in slow continuous pan mode, intermittent pan mode, 2-position patrol-mode PTZ, or zooming-in/zooming-out
Air Turbulence	Category includes outdoor videos showing air turbulence caused by rising heat.

Tabla 2.1: Categorias de video

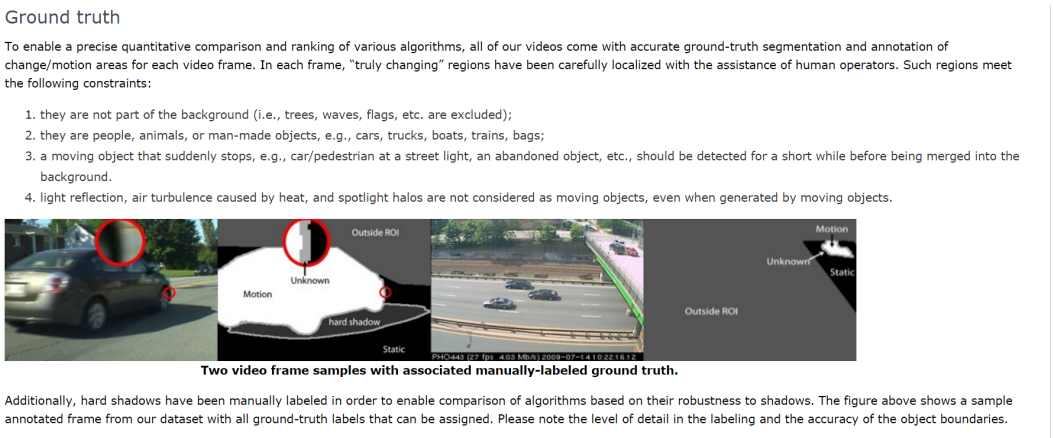


Figura 2.3: Ejemplo Frente Fondo

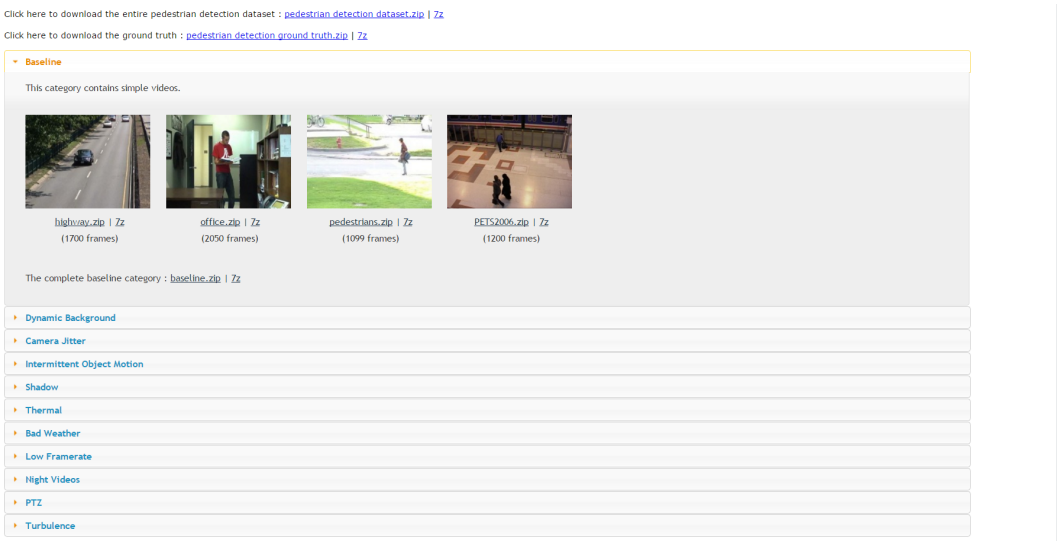


Figura 2.4: Descarga Datasets

enlaces de descarga a los datasets 2012 y 2014.

2.2.2. CDNET: UPLOAD

En este módulo se indican los pasos a seguir por el usuario para enviar sus resultados a la plataforma. Este módulo se divide en 2 submódulos, en el primer submódulo "UPLOAD FOR 2012 DATASET" se entregan los resultados para el DATASET 2012, en el segundo submódulo "UPLOAD FOR 2014 DATASET" se entregan los resultados para el DATASET 2014. Ambos submódulos son análogos, la unica diferencia entre ambos es el dataset que se tiene como referencia para evaluar los resultados.

En ambos submódulos, se facilita al usuario un ejemplo de como debe ser la entrega de resultados, es decir, estructura de carpeta y formato de archivos. Finalmente, se requiere cumplimentar una serie de campos tipo formulario con informacion personal del usuario y algoritmo a evaluar.

Adicionalemnte, se dispone del módulo "UTILITIES" que ofrece a los usuarios la posibilidad de evaluar sus algoritmos en un entorno local sin necesidad de enviar sus resultados a través de la plataforma para ser evaluados. Para realizar la evaluación en local, se proporciona los algoritmos de evaluación en formato Matlab o Python. De esta forma, el usuario puede evaluar sus distintos algoritmos con distintos parámetros e ir observando los resultados, este módulo reduce considerablemente la carga computacional en el lado del servidor de CDNET.

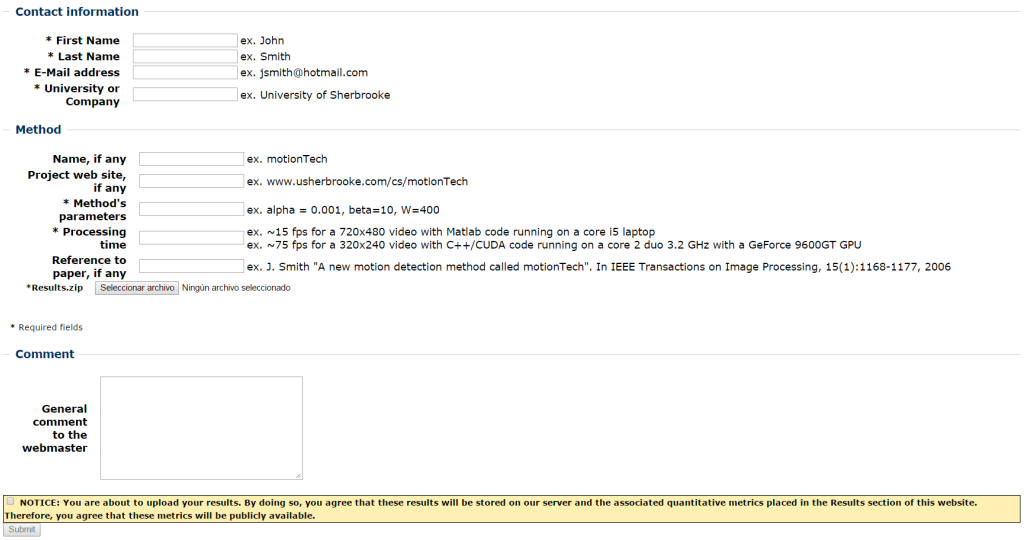
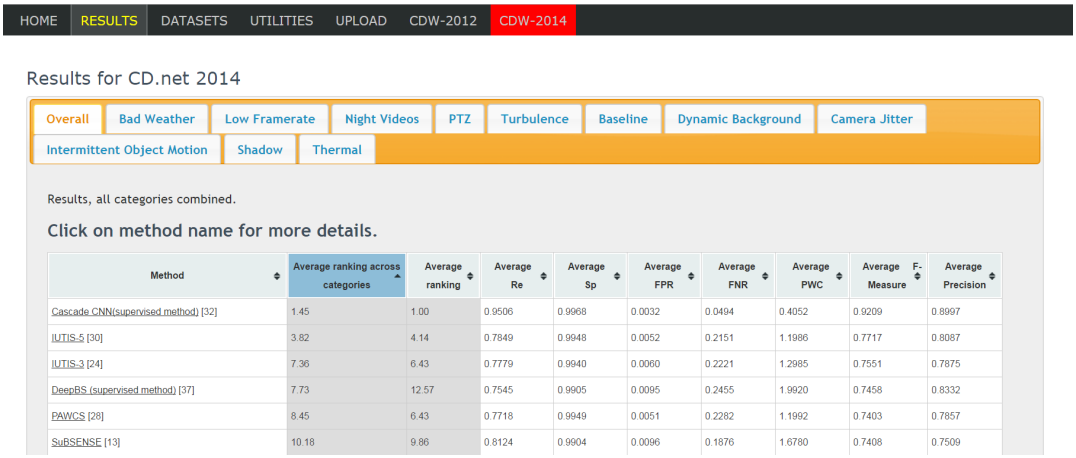


Figura 2.5: Formulario solicitud

2.2.3. CDNET: RESULTS

Finalmente, en este módulo se describe el sistema de métrica de evaluacion realizado sobre los resultados enviados por los usuarios. El sistema de métricas se basa en los elementos de la matriz de confusión: TP, FP, TN, FN.

- Average ranking accross categories : (sum of ranks for all categories) / (number of categories)
- Average ranking : (rank:Recall + rank:Spec + rank:FPR + rank:FNR + rank:PWC + rank:FMeasure + rank:Precision) / 7
- TP: True Positive
- FP: False Positive
- FN: False Negative
- TN: True Negative
- Re (Recall): TP / (TP + FN)
- Sp (Specificity): TN / (TN + FP)
- FPR (False Positive Rate): FP / (FP + TN)
- FNR (False Negative Rate): FN / (TP + FN)



Method	Average ranking across categories	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
Cascade CNN(supervised method) [32]	1.45	1.00	0.9506	0.9968	0.0032	0.0494	0.4052	0.9209	0.8997
IUTIS-5 [30]	3.82	4.14	0.7849	0.9948	0.0052	0.2151	1.1986	0.7717	0.8087
IUTIS-3 [24]	7.36	6.43	0.7779	0.9940	0.0060	0.2221	1.2985	0.7551	0.7875
DeepBS (supervised method) [37]	7.73	12.57	0.7545	0.9905	0.0095	0.2455	1.9920	0.7458	0.8332
PAWCS [28]	8.45	6.43	0.7718	0.9949	0.0051	0.2282	1.1992	0.7403	0.7857
SuBSENSE [13]	10.18	9.86	0.8124	0.9904	0.0096	0.1876	1.6780	0.7408	0.7509

Figura 2.6: Resultado Dataset 2014

- PWC (Percentage of Wrong Classifications): 100 * (FN + FP) / (TP + FN + FP + TN)
- F-Measure: (2 * Precision * Recall) / (Precision + Recall)
- Precision: TP / (TP + FP)
- FPR-S: Average False positive rate in hard shadow areas

Además, se muestran los resultados de la evaluación realizada para cada algoritmo. Los resultados que se muestran no son para cada video, se muestran resultados agregados del algoritmo para todos los videos y para cada categorizacion de video menciondas en la tabla 2.1. En la figura 2.6se muestra los resultados para algunos algoritmos.

Además, se habilita un hipervínculo por cada algoritmo que nos redirige a otra página donde se muestra la información asociada a dicho algoritmo como author, email, etc. En esta nueva página se dispone únicamente los elementos de la matriz de confusión por cada video evaluado. En la figura 2.7se muestra un ejemplo de la informacion y resultados mostrados para el algoritmo “Cascade CNN”.

Adicionalmente, se dispone de los módulos “CDW-2012” y “CDW-2014” donde se detallan los workshops realizados para divulgar la existencia de la plataforma. En los workshops se incluyen ponencias de equipos que han participado en el proyecto CD-NET, así como invitados especiales relacionados con tecnologías de visión artificial. En la figura 2.8se muestra el orden del día del Workshop realizado en 2014.

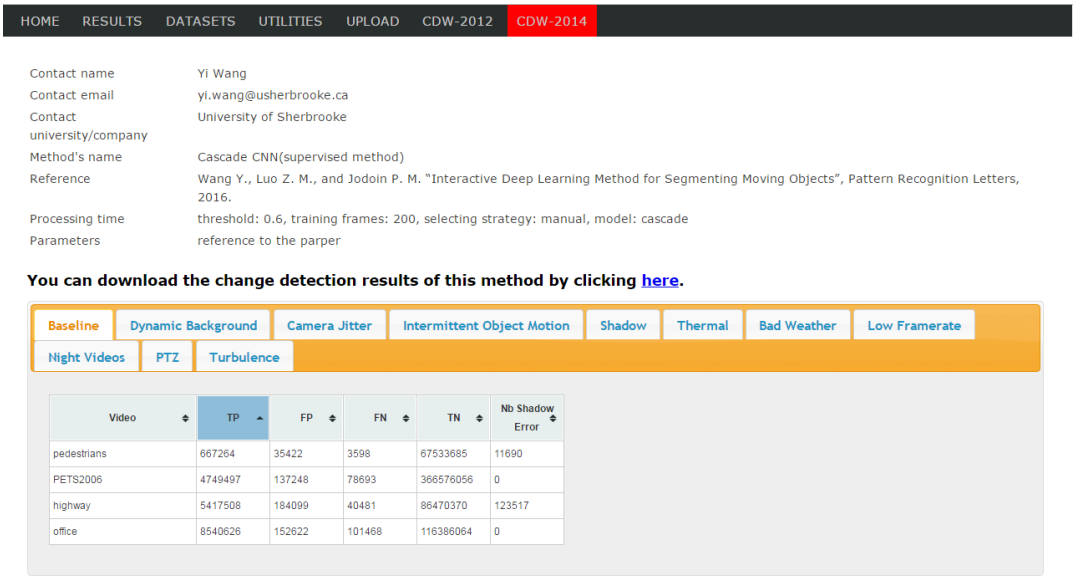


Figura 2.7: Resultado algoritmo Cascade CNN

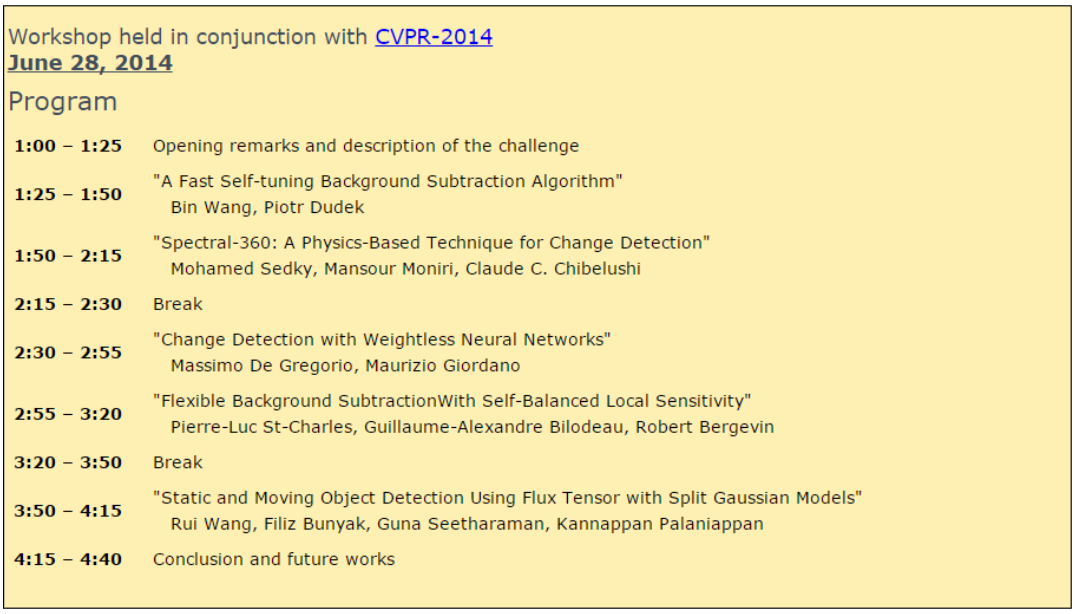


Figura 2.8: Workshop CDNET 2014

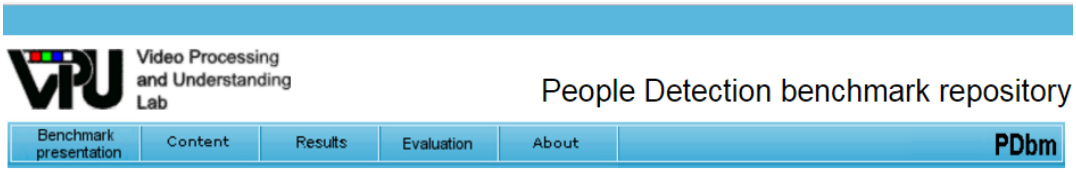


Figura 2.9: Módulos PDbm

2.3. People detection benchmark repository (PDbm)

En esta sección describiremos la plataforma web de evaluación People Detection Benchmark repository (PDbm). El objetivo principal de este proyecto es automatizar la plataforma PDbm basandonos en la plataforma CDNET.

La plataforma web PDbm sigue la filosofía de CDNET con la diferencia que evalúa algoritmos de detección de personas.Todos los videos del dataset que ofrece están orientados a este tipo de algoritmos. La plataforma CDNET se divide en 5 módulos. En la figura 2.9????? se muestran dichos módulos.

En el primer módulo "Benchmark presentation" se describe de forma resumida el objetivo de la plataforma, así como las personas que la organizan y mantienen. A continuación describiremos los módulos más relevantes de la plataforma.

2.3.1. PDbm: Content

En este módulo se enumeran los videos seleccionados por PDbm del "dataset" de la plataforma CDNET. Como hemos mencionado anteriormente, PDbm tiene como objetivo evaluar algoritmos de detección de personas, por ello todos los videos seleccionados del dataset de CDNET tienen personas en su contenido. Además, la plataforma PDbm define la caracteristica "Background complexity" a las categorias establecidas por CDNET y añade la característica "Classification complexity" con los valores LOW, MEDIUM, HIGH. Por último, se explica qué se considera "persona" en cada frame del video así como la métrica de evaluación que se usa para evaluar los resultados. La métrica de evaluación utilizada en PDbm es el área bajo la curva Precission-Recall. En la figura 2.10 se muestra algunos videos del dataset PDbm.

2.3.2. PDbm: Evaluation

En este módulo se realizan las entregas de los resultados. Cada usuario envía los resultados obtenidos por su algoritmo de deteccion de personas. También se especifica la estructura y el formato de los ficheros de entrega. Actualmente, se debe enviar un email a la direccion pdbm-l@uam.es. Desde el VPULab, existe una persona encargada

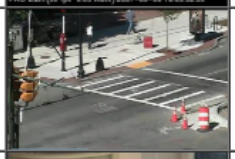
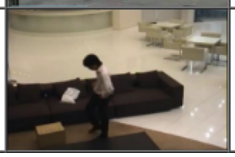
	Video		Background complexity	Classification complexity	#Frames
1		office	Baseline	Low	2050
2		pedestrians	Baseline	Low	1099
3		PETS2006	Baseline	Medium	1200
4		fall	Dynamic Background	High	4000
5		overpass	Dynamic Background	Medium	3000
6		badminton	Camera Jitter	Medium	1150
7		sidewalk	Camera Jitter	High	1200
8		abandonedBox	Intermittent Object Motion	High	4500
9		sofa	Intermittent Object Motion	Low	2750

Figura 2.10: Dataset PDbm

Video	HOG[1]	ISM[2]	Fusion[3]	Edge[4]	DTDP[5]	ACF[6] Inria	ACF[6] Caltech	Faster-RCNN[7] VOC0712_ZF	Faster-RCNN[7] VOC0712_VGG
1	89.3	71.4	34.9	84.5	96.7	99.3	86.4	**99.8**	99.7
2	63.2	82.9	92.5	90.2	66.3	77.1	92.9	80.8	**98.2**
3	55.6	75.7	64.3	71.7	69.7	68.9	56.0	78.0	**82.9**
4	10.1	1.0	0.5	5.4	13.2	33.9	**59.1**	20.8	37.5
5	61.3	71.2	3.5	7.1	85.1	51.6	24.7	**96.3**	96.2
6	49.9	34.6	8.1	31.7	74.9	67.2	54.4	88.4	**93.1**
7	0.0	3.0	7.6	7.2	11.0	2.7	**89.4**	70.1	75.9
8	0.0	12.2	17.7	21.4	0.0	4.6	33.4	59.0	**79.8**
9	47.4	65.9	45.9	74.4	90.4	72.2	76.1	93.1	**94.7**
10	7.2	5.5	21.2	14.3	0.7	8.7	59.5	79.4	**91.0**
11	10.7	5.9	2.1	33.7	11.4	8.6	34.8	44.8	**95.4**
12	82.0	76.2	65.7	70.5	92.4	91.1	92.6	98.6	**99.6**
13	70.9	73.6	15.5	59.3	80.2	87.2	81.6	93.3	**98.6**
14	13.6	29.2	12.7	46.9	44.1	21.3	51.1	85.7	**95.9**
15	46.5	20.5	16.6	41.2	67.9	70.0	86.2	97.4	**98.5**
16	21.1	71.5	37.7	60.9	83.3	89.6	54.5	94.3	**95.9**
Average AUC	39.3	43.8	27.9	45.0	55.4	53.4	64.5	80.0	**90.0**

Tabla 2.2: PDbm resultados video

de revisar esta bandeja periódicamente y evaluar los resultados enviados, así como recoger los datos del usuario. El objetivo del proyecto es trabajar fundamentalmente en este módulo, donde se automatizará la recogida de resultados y los datos asociados al usuario.

2.3.3. PDbm: Results

En este módulo se muestran los resultados de la evaluación realizada para cada algoritmo en forma matricial. Se muestran los algoritmos por columnas y los resultados de evaluación de cada video por filas. Se disponen de 3 tipos de tablas de resultados. En la primera tabla 2.2 se muestran los resultados por cada vídeo. En la segunda tabla 2.3 se muestran los resultados por cada categoria “Background complexity”. En la última tabla 2.4 se muestran los resultados por cada categoria “Classification complexity”.

Por último, se muestra un hipervínculo por cada algoritmo que nos redirige a otra página donde se muestra la información asociada a dicho algoritmo al igual que el módulo “RESULTS” de la plataforma CDNET. En esta nueva página se dispone también de 3 tablas análogas a las mostradas en la subsección anterior.

Background complexity	HOG[1]	ISM[2]	Fusion[3]	Edge[4]	DTDP[5]	ACF[6] Inria	ACF[6] Caltech	Faster-RCNN[7] VOC0712_ZF	Faster-RCNN[7] VOC0712_VGG
Baseline	69.4	76.7	63.9	82.1	77.6	81.8	78.4	86.2	**93.6**
Dynamic Background	35.7	36.1	2.0	6.3	49.2	42.8	41.9	58.5	**66.9**
Camera Jitter	25.0	18.8	7.9	19.4	42.9	34.9	71.9	79.2	**84.5**
Intermittent Object Motion	16.3	22.4	21.7	35.9	25.6	23.5	51.0	69.1	**90.2**
Shadow	46.8	54.2	29.6	55.8	73.6	71.8	73.2	93.8	**97.7**
Average AUC	38.6	41.6	25.0	39.9	53.8	51.0	63.3	77.4	**86.6**

Tabla 2.3: PDbm resultados background complexity

Classification complexity	HOG[1]	ISM[2]	Fusion[3]	Edge[4]	DTDP[5]	ACF[6] Inria	ACF[6] Caltech	Faster-RCNN[7] VOC0712_ZF	Faster-RCNN[7] VOC0712_VGG
Low	62.3	73.6	48.7	73.3	84.9	86.1	80.7	93.3	**97.8**
Medium	53.3	50.5	23.1	37.9	74.4	64.4	55.3	90.0	**92.7**
High	6.9	9.5	10.3	21.5	13.4	13.3	54.5	60.0	**79.2**
Average AUC	40.8	44.5	27.4	44.2	57.6	54.6	63.5	81.1	**89.9**

Tabla 2.4: PDbm resultados classification complexity

2.4. Conclusiones

En esta sección se detallará las conslusiones a las que se ha llegado para realizar el proyecto partiendo de las plataformas de evaluacion de algoritmos existentes como ChangeDetection.net o People Detection Benchmark repository.

Se observa que la plataforma CDNET se basa un sistema simple y preciso. Se centra en 3 procesos: Recogida de datos, procesado de datos y publicacion de resultados. La plataforma PDbm también sigue este tipo de sistema pero no es automática. Por ello, se modificará la plataforma PDbm siguiendo el modelo de recogida de datos en forma de formulario mostrado en el módulo “UPLOAD” de CDNET, pero se respetará la forma de mostrar los datos en el módulo “Results” de PDbm debido a que el objetivo principal de este proyecto es automatizar la plataforma PDbm. Por ello, no se realizarán desarrollos en la interfaz con el usuario ya que como hemos mencionado, el objetivo principal es automatizar la plataforma PDbm .Por último, se desarrollará un modulo extra de administrador en PDbm para gestionar la plataforma de manera ágil y sencilla.

Capítulo 3

Diseño

En este capítulo se describe el diseño global del sistema de evaluación automática que se desea realizar sobre la plataforma estática PDbm. El sistema se divide en 4 bloques funcionales independientes. Cada uno de estos bloques es modulable siempre y cuando mantenga únicamente su utilidad final de aplicación. En la sección 3.2 describiremos el bloque correspondiente al servidor web encargado de procesar las peticiones HTTP de los usuarios y recibir sus resultados, así como el modelo de desarrollo que deberá seguir el sistema. Después, en la sección 3.3 hablaremos del bloque correspondiente a la base de datos donde almacenaremos la información asociada a los algoritmos enviados por los usuarios y los resultados evaluados por la plataforma. Más adelante, en la sección 3.4 describimos el bloque encargado de evaluar los resultados enviados por el usuario. Por último, en la sección 3.5 se detallará el bloque correspondiente al gestor de correo. Este gestor se encarga de notificar al administrador de la plataforma de nuevos registros de algoritmos, entre otras notificaciones.

3.1. Introducción

Como se ha mencioando en el capítulo anterior, el desarrollo de algoritmos de deteccion o reconocimiento está creciendo cada vez más. De manera similar, el desarrollo de plataformas web tambien ha notado un gran crecimiento en los últimos años. Exiten múltiples formas de diseñar el sistema de una plataforma web en funciñon de la utilidad final. Por ejemplo existen plataformas didácticas cuya finalidad es brindar y almacenar material educativo, además de interactuar las opiniones de los usuarios en un foro, entre otras funciones. Un ejemplo extendido de este tipo de plataforma es Moodle, utilizada en la Universidad Autónoma de Madrid. Por otro lado, también existen plataformas de publicación cuya finalidad es brindar y almacenar contenido

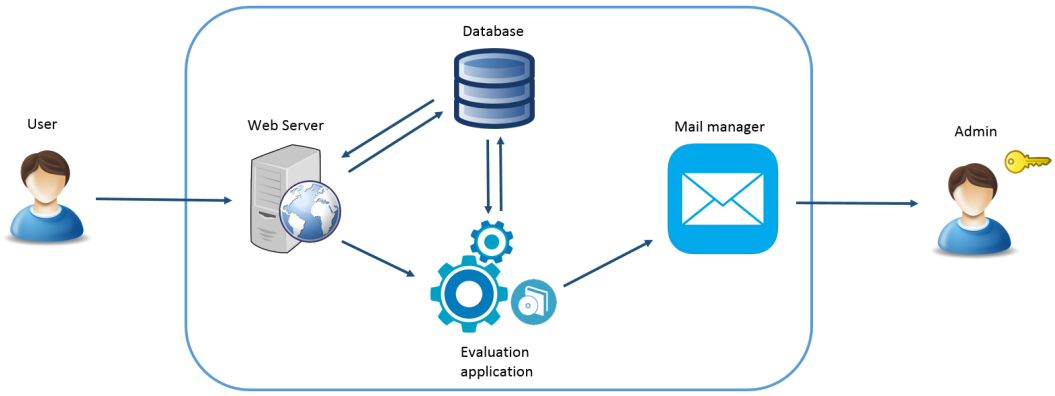


Figura 3.1: Sistema automático PDbm

audiovisual, además de sugerir contenidos personalizados a los usuarios. Un ejemplo extendido de este tipo de plataforma es Youtube. Existen muchos más tipos de plataformas extendidas actualmente con multitud de funcionalidades. Las funcionalidad mas básicas que brindan la mayoría de plataformas es el almacenamiento y visualización de datos, contenidos, etc. En este proyecto se desarrollo dichas funcionalidades y se añadirá otra, el procesado de datos. En PDbm el procesado de datos se traduce en evaluación de resultados.

En la figura 3.1 se muestra el diseño del sistema automático propuesto para la plataforma PDbm.

A continuación se describirá cada uno de los módulos mostrados en la figura 3.1.

3.2. Servidor Web y modelo de desarrollo

En esta sección describiremos el primer bloque del sistema propuesto. El elemento principal para cualquier plataforma web es disponer de un servidor web que procese las peticiones HTTP de los usuarios. En este servidor se almacenarán los scripts encargados de ejecutar el sistema completo. En otras palabras, este bloque es el motor del sistema. En cuanto a la organización de los scripts,se adoptará la programación por capas como modelo de desarrollo. En la figura 3.2 se muestra el modelo de programación por capas.

El modelo de programación por capas consta de 3 capas. La primera, es la capa de presentación, donde se encuentran los scripts que generan la interfaz con el usuario. La segunda, es la capa lógica de negocio, donde se encuentran los scripts que generan los procesos de la plataforma. Esta capa se comunica con la capa de presentaciñon para recibir las solicitudes y presentar los resultados tras ejectuar, si es necesario,

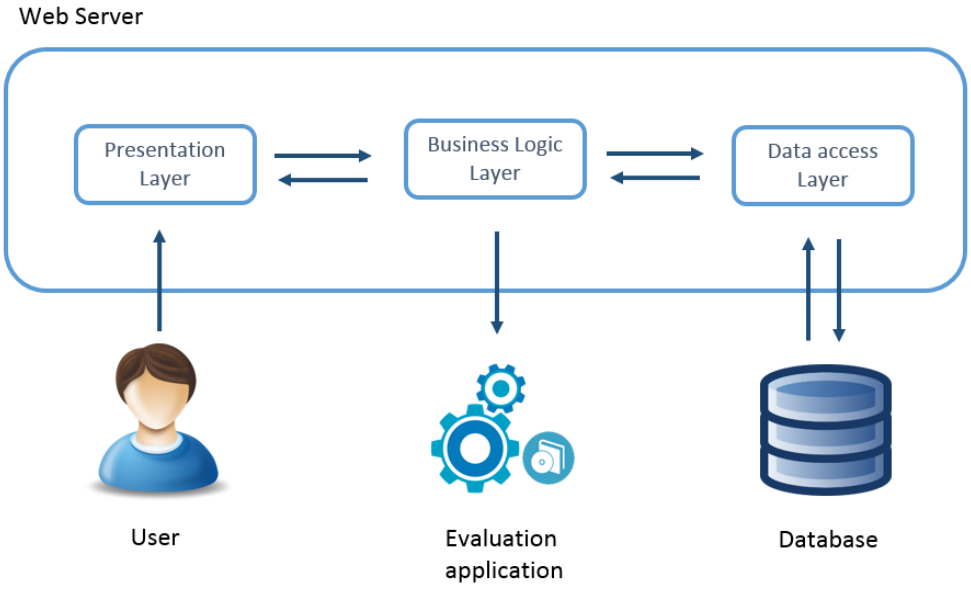


Figura 3.2: Modelo de programación por capas

los procesos correspondientes a la solicitud. Por último, se encuentra la tercera capa de datos. Esta capa contiene los scripts que realizan la comunicación con la base de datos de la plataforma. Estas comunicaciones pueden ser de lectura o escritura.

En conlusión se ha adoptado este modelo de programación para realizar el desarrollo en varios niveles. En caso de un cambio en la plataforma, no se tendrá que revisar todos los scripts del sistema, si no únicamente los de la capa afectada. De esta manera, diseñamos un sistema modulable, escalable y mantenible de forma ágil y sencilla.

3.3. Base de datos

En esta sección describiremos el segundo bloque funcional del sistema propuesto. En primer lugar, el almacenamiento y la gestión de datos son procesos fundamentales en el desarrollo de una plataforma de cualquier tipo, por esta razón se le dedica un bloque funcional independiente. Además, actualmente estos procesos son una ciencia que evoluciona cada vez más debido al incremento abrupto de datos en los últimos años. Por lo tanto, en caso de tener que cambiar de gestor de base de datos debido a una evolución de esta tecnología, se realizaría una migración de los datos y una adaptación en la comunicación de la capa de datos con el nuevo gestor. Por otro lado, resulta útil tener aislado este bloque para realizar un correcto modelado de

datos en función del objetivo de la plataforma. De esta manera, nos abstraemos del desarrollo de scripts en el servidor web y realizamos un modelo de datos más eficiente. En coclusión, se requiere un bloque funcional dedicado al almacenamiento, gestión y modelado de datos.

3.4. Aplicación de evaluacion

En esta sección describiremos el tercer bloque funcional del sistema propuesto. Uno de los procesos mas costosos computacionalmente en la plataforma PDbm es realizar la evaluación de resultados. Este tipo de procesos, al igual que muchos procesos de algoritmia, requieren de una cantidad considerable de operaciones matemáticas. Por esta razon, existen sistemas de software matemático diseñados específicamente para dicho fin. Por lo tanto, en el sistema propuesto, se requiere una aplicación externa al servidor web para realizar la evaluación de resultados de manera ágil.

3.5. Gestor de comunicaciones

En esta última sección describiremos el cuarto bloque funcional del sistema propuesto. En cualquier tipo de plataforma que disponga de interactividad con el usuario, existe una serie de comunicaciones entre la plataforma y el usuario. Estas comunicaciones pueden ser totalmente asíncronas y manuales por parte del gestor de la plataforma, quien tendrá la obligación de revisar mensualmente la actividad. En este caso, se requiere un gestión de notificaciones semi-automática. Existen determinadas comunicaciones con el usuario que no pueden ser automáticas ya que se requiere de sentido humano para atender distintas dudas de los usuarios. Por otro lado, también existen comunicaciones que se pueden automatizar como eventos de inicio o de fin.

3.6. Conclusiones

En esta sección se detallarán las conclusiones a las que se ha llegado tras proponer un diseño del sistema automático para la plataforma PDbm. En primer lugar, se ha demostrado que se puede dividir el sistema en bloques para hacerlo modulable, escalable y flexible. Estas características resultan indispensables para plataformas que evolucionan constantemente debido a su naturaleza, en este caso evaluación de algoritmos de detección de personas. En segundo lugar, para cada bloque se dispone de multitud de tecnologías ya desarrolladas y puestas en producción en múltiples plataformas. En conclusión, se ha diseñado un sistema que se adapta a las necesidades

del objetivo del proyecto, y además, es sencillo de desarrollar, gestionar y poner en producción.

Capítulo 4

Desarrollo

En este capítulo de explicarán las tecnologías y los desarrollos utilizados para implementar el diseño del sistema automático propuesto en el capítulo anterior. Debido al diseño propuesto en bloques, existe la posibilidad de trabajar con distintas tecnologías de manera independiente y unirlas para que trabajen como un sistema único. En la sección 4.2 se describirá el entorno de desarrollo web WampServer utilizado para los bloques 1 y 2 de la sistema propuesto en la figura 3.1. Después, en la sección 4.3 se explicará cómo se ha desarrollado el modelo de capas propuesto en la figura 3.2. Además, también se detallará el paradigma de programación utilizado en cada capa del modelo. Más adelante, en la sección 4.4se detallará el uso que se hace del software estadístico Matlab correspondiente tercer bloque del sistema diseñado. Por último, en la sección **??** se describirá las comunicaciones manuales y automáticas que se realizarán con la herramienta de Microsoft Office, Outlook, para gestionar el correo. Esta última seccion se corresponde al bloque 4 del sistema propuesto.

4.1. Introducción

Actualmente existe una cantidad voluminosa de distintas tecnologías, API's, frameworks y librerias de uso público a disposición de cualquier persona. En este proyecto se han valorado distintas tecnologías que puedan adaptarse al sistema propuesto. Finalmente se han elegido aplicaciones, entornos y modelos que cumplen los requisitos, y a su vez se dispone de un conocimiento básico de ellos. En la figura 4.1 se muestra el desarrollo específico para cada bloque del sistema propuesto en la figura 3.1.

A continuación se describirá cada una de las tecnologías mostradas en la figura 4.1.

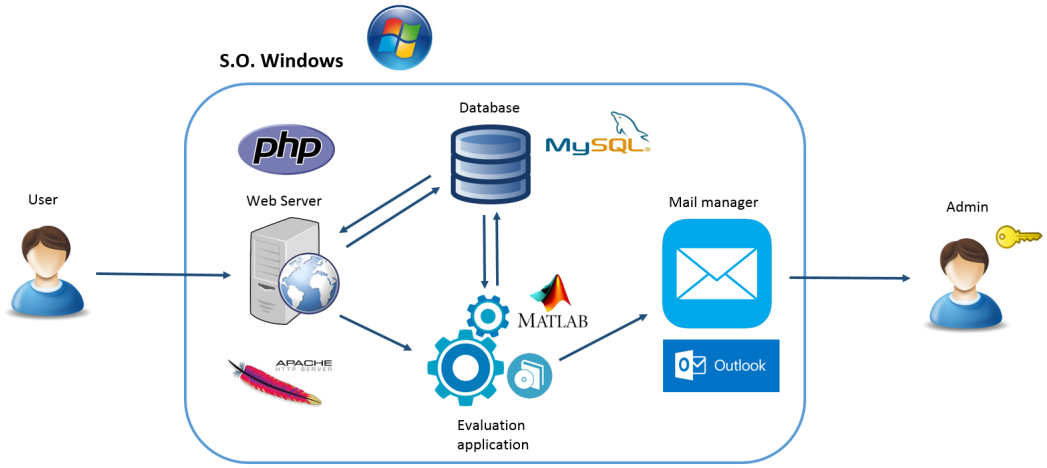


Figura 4.1: Desarrollo sistema PDbm

4.2. WampServer (Apache, PHP, MySQL on Windows)

En esta sección se describirá la tecnología base que se ha utilizado para desarrollar el sistema automático PDbm. Se ha utilizado WampServer como entorno de desarrollo web. WAMP se basa en un conjunto de herramientas: Windows como sistema operativo, Apache[3] como servidor web, MySQL[4] como gestor de bases de datos y PHP[5], Perl, o Python como lenguajes de programación. Este entorno de desarrollo aprovisiona de tecnologías para los bloques 1 y 2 de la figura 3.1. En primer lugar, Windows es uno de los sistemas operativos más extendidos actualmente, sencillo de gestionar y utilizado en la universidad Autónoma de Madrid, donde se esta realizando el proyecto. En segundo lugar, Apache HTTP server es un servidor web de código abierto, disponible para la mayoria de versiones de Windows. Es sencillo de instalar, configurar y poner en producción. En tercer lugar, MySQL es uno de los sistemas de gestión de bases de datos relacional más extendidos para entornos de desarrollo web. Además dispone del estándar ODBC para poder conectarse a cualquier aplicación que soporte dicho estándar, en general se desarrollan este tipo de drivers para la mayoría de aplicaciones. Cabe mencionar que tamién se incluye phpMyAdmin[6] en WAMP[7] como herramienta que facilita la gestion de MySQL. Por último, se dispone de PHP como lenguaje de programación. PHP es un lenguaje diseñado para el desarrollo web del lado del servidor. En este proyecto encaja perfectamente con el sistema diseñado en 3.1. Asimismo, este lenguaje se puede integrar de manera sencilla con HTML. Los scripts que se disponen de la plataforma PDbm son ficheros HTML.

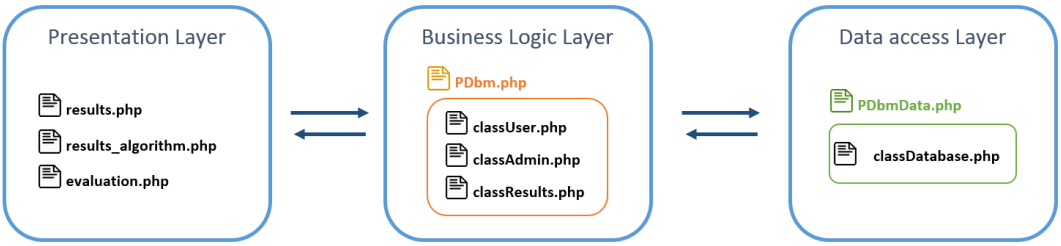


Figura 4.2: Desarrollo modelo de capas

4.3. Desarrollo de capas y estructura de tablas

En esta sección se detallará los scripts desarrollados en cada capa del modelo de capas propuesto en la figura 3.2. Posteriormente se explicará la estructura de tablas adoptada en la base de datos MySQL para gestionar la plataforma.

En la figura 3.2 se muestran los distintos scripts desarrollados para cada capa. El paradigma de programación utilizado ha sido programación orientada a objetos y se ha desarrollado un fichero PHP por cada clase.

A continuación describiremos de forma general los scripts PHP de cada cada:

- Presentation Layer: En esta capa se ha desarrollado el fichero "evaluation.php" como una extensión del módulo "evaluation" descrito en 2.3.2. Se añaden los formularios de tipo texto para enviar la información del usuario y un botón para adjuntar los resultados a evaluar. Por otro lado, también se han desarrollado los fichero "results.php" y "results_algorithm.php" correspondiente al módulo "results" descrito en 2.3.3. Estos ficheros muestran los resultados de los algoritmos visibles elegidos por el administrador de la plataforma de forma dinámica.

- Business Logic Layer: En esta capa se ha desarrollado las clases orientadas a las entidades que realizan las acciones principales. Por ejemplo, el fichero"classUser.php" contiene las funciones necesarias para cargar la base de datos con la información del nuevo usuario, comunicarse con la aplicación de evaluación para que inicie su proceso, etc. En el caso del fichero "classAdmin.php", contiene las funciones necesarias para eliminar un algoritmo del sistema, dar visibilidad a un algortimo almacenado, etc. Estos ficheros contienen las funciones que realizan los procesos del sistema.

- Data acces Layer: En esta capa se han desarrollado las clases orientadas a la conexión con las distintas fuentes de datos que existan en el sistema. En este

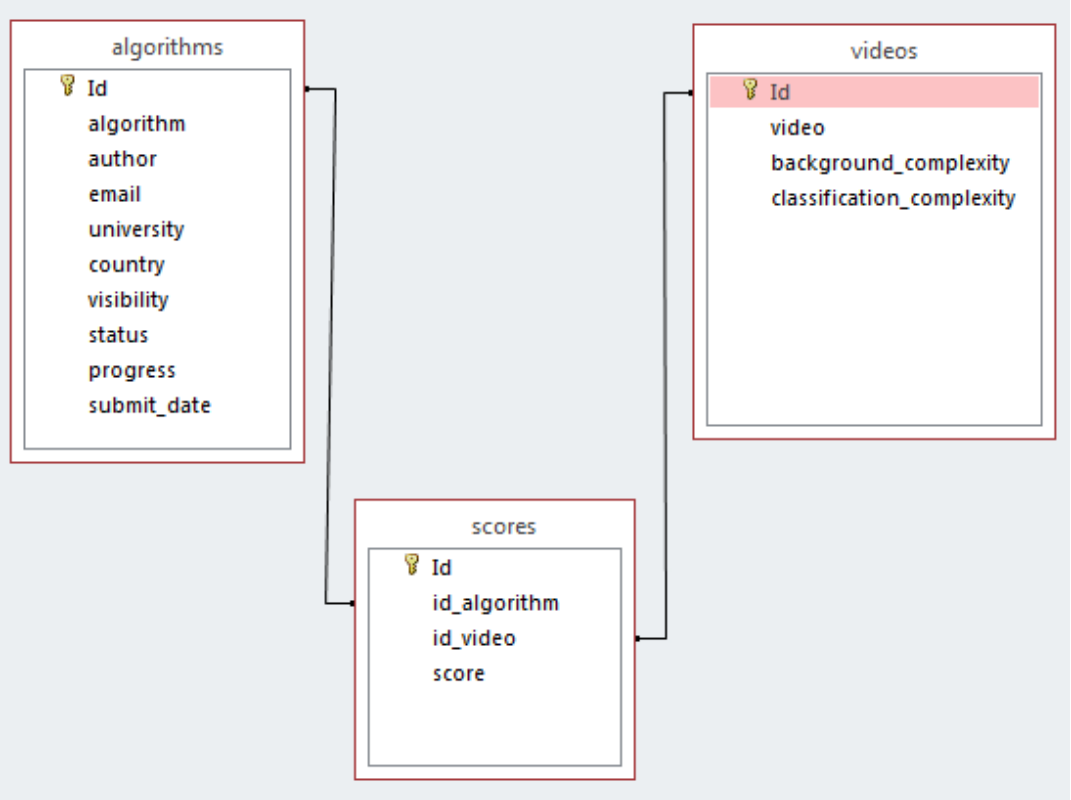


Figura 4.3: Estructura de tablas

caso, se dispone únicamente de un fichero que se encarga de realizar la conexión con MySQL.

Existen múltiples formas de desarrollar un sistema de ficheros. En este proyecto se ha tomado la filosofía descrita anteriormente para poder integrar ficheros de forma ágil. A continuación, en la figura 4.3 se muestra la estructura de tablas diseñada en MySQL.

La base de datos se llama PDbm y contiene 3 tablas: algorithms, videos, scores.

- algorithms: Contiene un registro por cada envío de resultados para un algoritmo. El campo "status" puede tomar los valores: "pending" si el algoritmo esta pendiente de evaluar,"in process" si el algoritmo esta siendo evaluado, "ready" si se ha terminado de evaluar el algoritmo. Después, el campo "progress" indica la cantidad de progreso en la evaluacion de los resultados que se tiene hasta el momento en una escala del 1 al 100. Posteriormente, el campo "visibility" indica si mostramos los resultados del algoritmo de forma pública a través de la web con el previo consentiemiento del autor. Este campo es de tipo booleano.

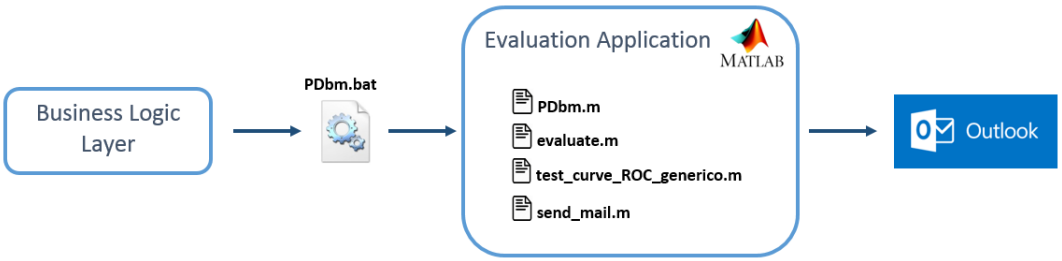


Figura 4.4: Matlab y Outlook

Finalmente, el resto de campo contienen información adicional del algoritmo a evaluar.

- videos: Contiene un registro por cada video del dataset a evaluar. Los campos indican las distintas categorias mencionadas en 2.3.1.
- scores: Contiene un registro para cada puntuación del algoritmo para cada video del dataset.

Como se puede ver en la figura 4.3, se dispone de 2 relaciones entre las tablas "algorithms"-"scores" y las tablas "videos"-"scores" a través de las claves primarias de las tablas "algorithms" y "videos". De esta manera se mejora considerablemente la gestión y comprensión de los datos.

4.4. Matlab y Outlook

En esta sección explicará la integración de la aplicación Matlab[8] como aplicación de evaluación y Outlook como gestor de correo.

En la figura 4.4 se muestra el subsistema encargado de evaluar los resultados de los usuarios y a su vez notificar al administrador de la actividad de la plataforma.

Como se ha mencionado en 4.3, la capa lógica de negocio es la encargada de realizar los procesos de la plataforma. Esta capa se comunica con la aplicación de evaluacion, en este caso Matlab, através de un fichero BAT. De esta manera, se consigue ejecutar los procesos de evaluación.

En la aplicacion de evaluación se ha modificado ligeramente el script "test_curve_ROC_generico.m" proporcionado por VPULab para integrarlo en el sistema automático PDbm. Este script obtiene el area bajo la curva Precission-Recall utilizado como métrica de evaluación para un algoritmo y un video del dataset. Posteriormente, se han desarrollado los scripts "PDbm.m" y "evaluate.m" para modular la aplicación del sistema y poder integrar cualquier otro algoritmo evaluador. Finalmente, se desarrolla el script

"send_mail" que se comunica con el gestor de mail, en este caso Outlook, para enviar notificaciones al administrador cuando se inicia o finaliza un proceso de evaluación.

Capítulo 5

Resultados

En este capitulo se mostrará los resultados obtenidos tras realizar los desarrollos mencionados en el capítulo anterior. En la sección 5.1 se mostrará la interfaz desarrollada para que el usuario pueda participar en la plataforma. Posteriormente, en la sección 5.2 se mostrará el modulo de control desarrollado para el administrador de la plataforma.

5.1. Interacción usuario

En la figura 5.1, se muestra enarmcada en rojo la zona donde se ha añadido el formulario al módulo PDbm para que el usuario pueda enviar sus datos y resultados.

El proceso completo de la plataforma se inicia cuando el usuario hace click en el botón "Send". Tanto al iniciar como el terminar el proceso, se envía una notificación al administrador para que sea consciente de la entrada de un nuevo algoritmo.

Posteriormente, las tablas del módulo "Result" mencionadas en 2.3.3 no sufren cambios importantes o de concepto.

5.2. Interacción administrador

En la figura 5.2se muestra el módulo de control del administrador con datos de ejemplo.

El administrador tiene disponible dos acciones: Actualizar y eliminar registros de algoritmos.

- Actualizar: Todos los algoritmos enviados tienen el campo "visibility" con valor FALSE por defecto. El administrador se encargará de poner dicho campo a TRUE tras acordar con el autor la publicación de sus resultados.

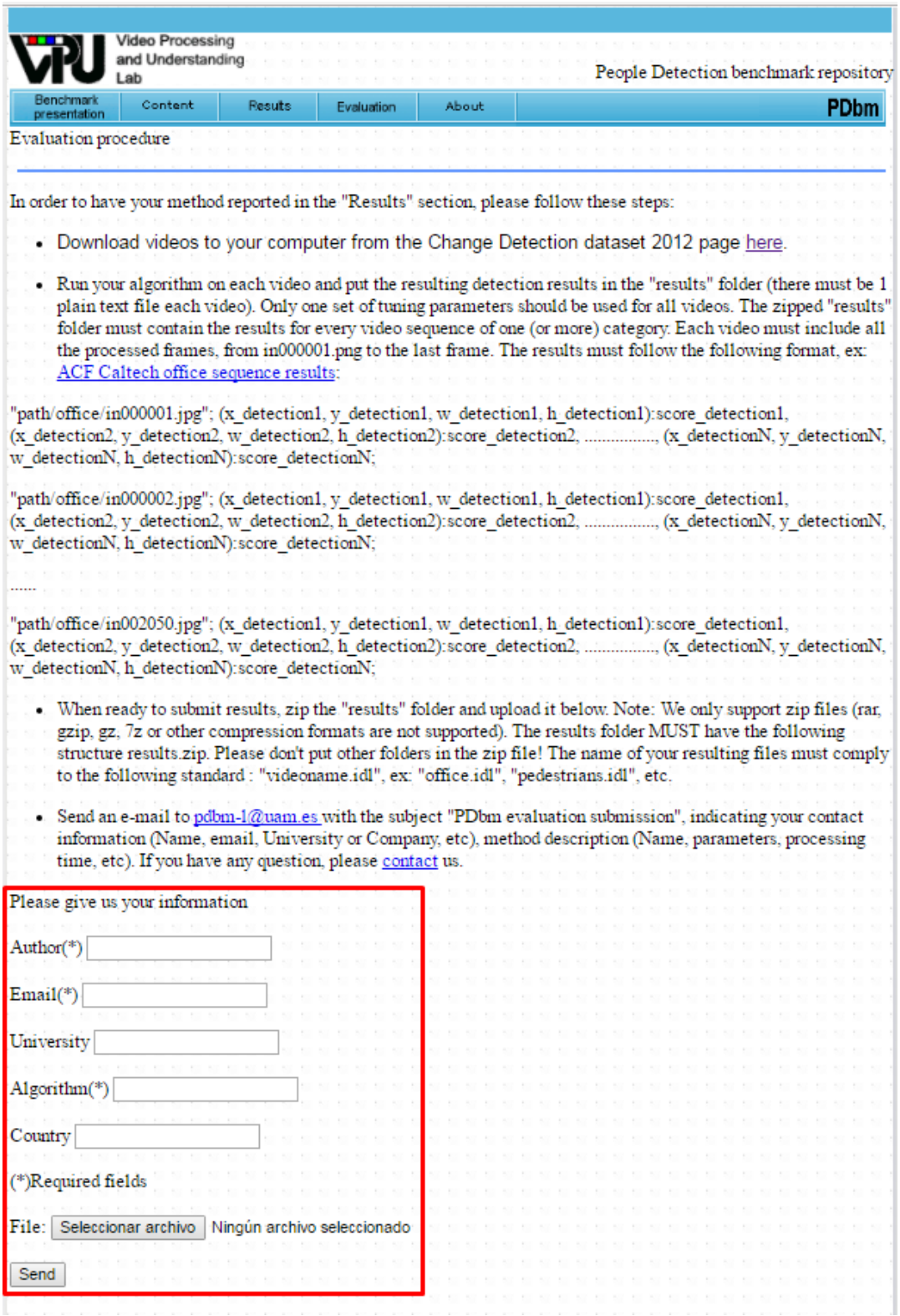
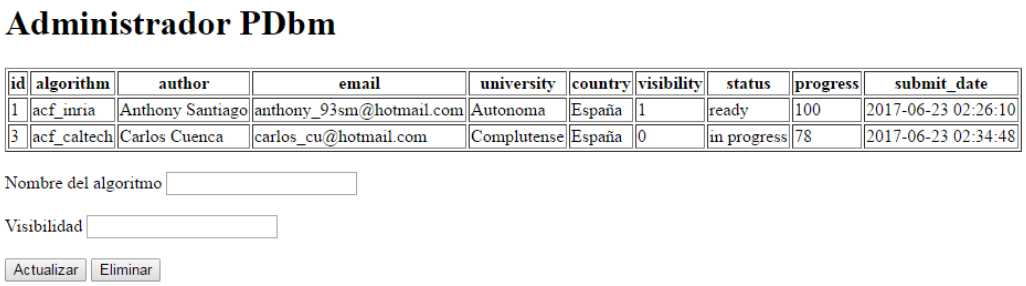


Figura 5.1: Resultado PDbm Evaluation



Administrador PDbm

id	algorithm	author	email	university	country	visibility	status	progress	submit_date
1	acf_inria	Anthony Santiago	anthony_93sm@hotmail.com	Autonoma	España	1	ready	100	2017-06-23 02:26:10
3	acf_caltech	Carlos Cuenca	carlos_cu@hotmail.com	Complutense	España	0	in progress	78	2017-06-23 02:34:48

Nombre del algoritmo

Visibilidad

Actualizar Eliminar

Figura 5.2: Resultado PDbm admin

- Eliminar: En caso de tener registros con campos erróneos o haya ocurrido algún problema en el proceso de evaluación de resultados, se dispone la opción de eliminar el algoritmo totalmente del sistema con este botón.

Capítulo 6

Conclusiones y trabajo futuro

En este último capítulo se expondrán las conclusiones a las que se ha llegado tras obtener los resultados de la plataforma PDbm desarrollada en el capítulo 4 a partir del sistema propuesto en el capitulo 3. Después, se detallará el trabajo futuro a realizar en la plataforma tras haber detectado los puntos donde se considera razonable poder continuar los desarrollos.

6.1. Conclusiones

En este proyecto se ha estudiado las plataformas CDNET y PDbm mencionadas en el capítulo 2. Ambas plataformas persiguen el mismo objetivo, evaluar algoritmos, pero en este proyecto se desea automatizar la plataforma PDbm, la cual esta orientada a evaluar algoritmos de detección de personas. Durante todo el proceso de automatización de PDbm se ha tomado como referencia la plataforma CDNET. En el estudio realizado de la plataforma CDNET, se ha observado que cuenta con muchos tipos de funcionalidades, principales y secundarias. En el diseño realizado en el capítulo 3 se ha cubierto las funcionalidades principales de CDNET y se han adaptado a la plataforma PDbm. Además, se ha propuesto un sistema de bloques, donde cada bloque contiene una finalidad específica. El motivo principal ha sido poder desarrollar cada bloque de forma independiente y abstraerse del resto de módulos en cierta medida. Otro de los motivos principales del diseño en bloques ha sido tener la capacidad desarrollar la plataforma de forma modular en el futuro. Posteriormente, se realizó un segundo estudio de la posibles tecnologías, aplicaciones que podrían encajar en el sistema propuesto. Tras ello, en el capítulo 4 se implementó cada bloque con tecnologías como WAMP, Matlab, Outlook, entre otras. Además, también se desarrolló un modulo de control para el administrador de la plataforma mostrado en el capítulo 5.

Gracias al diseño modular, cada tecnología se integro en su bloque correspondiente con facilidad. La mayor parte de la dificultad tuvo lugar en la comunicación entre los distintos bloques. Finalmente, se lograron los objetivos planteados en el capítulo 1, diseñar e implementar una plataforma de evaluación y comparación automátcia de algoritmos de detección de persona. Sin embargo, durante el desarrollo del proyecto, se localizaron diferentes aspectos a desarrollar en un nivel mas detallado en algunos bloques del sistema. A continuación, en la sección 6.2 se describirán dichos aspectos.

6.2. Trabajo futuro

Patiendo del trabajo realizado y los métodos adoptados para su desarrollo. Se puede tomar la figura 4.1 como referencia para lozalicar los aspectos a evaluar para su posterior desarrollo.

- En el primer bloque se ha adoptado el modelo de capas como metodología de desarrollo Con respecto a los scripts de la capa de visualización, únicamente se ha añadido las líneas de código necesarias para automatizar las solicitudes de evaluación y mostrar los resultados de evaluación de forma dinámica. El diseño estético web no ha sido modificado en ningun aspecto. Actualmente, existen librerías open-source, por ejemplo “node.js”, dedicadas exclusivamente a la innovación estética de páginas web. Resulta interesante indagar en este aspecto debido a la tendencia que está teniendo este tipo de desarrollos actualmente.

- En el segundo bloque, la base de datos utilizada ha sido MySQL. Se ha utilizado una estructura de tablas sencilla que disponga lo esencial para que el sistema propuesto funcione correctamente. No obstante, la gestión de datos también es otra ciencia que está evolucionando constantemente debido al crecimiento exponencial de volumen de datos. Otra posible rama de desarrollo sería la optimización del almacenamiento y rendimiento de la base de datos.

- En el tercer bloque se ha utilizado el software estadístico Matlab. Sin embargo, también existen aplicaciones de software matemático open-source, por ejemplo “python” o “R”. Este tipo de aplicaciones se pueden integrar en la plataforma y a su vez optimizar los algoritmos de evaluación para reducir coste computacional sobre el servidor.

- En el cuarto bloque, el gestor de correo utilizado ha sido Outlook. Se ha automatizado las notificaciones necesarias para que el administrador de la plataforma no tenga que revisarla periódicamente. Evidentemente, existen comunicaciones

entre la plataforma y el usuario que no se pueden automatizar y requieren el uso exclusivo de una persona humana. En este bloque no se detecta aspectos relevantes de mejora.

Otro aspecto general que no se ha profundizado durante el desarrollo del proyecto ha sido la seguridad informática. En los últimos años, muchas entidades han sufrido graves ataques en sus sistemas, plataformas, etc. debido a la falta de interés en este aspecto. Por lo tanto, cabe mencionar el estudio de seguridad en la plataforma como trabajo futuro.

Bibliografía

[1] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, June 2012.

[2] A. García-Martín, B. Alcedo, and J. M. Martínez, "Pdbm: people detection benchmark repository," *Electronics Letters*, vol. 51, no. 7, pp. 559–560, 2015.

[3] "Apache http server project." `https://httpd.apache.org/`.

[4] "Mysql." `https://www.mysql.com/`.

[5] "Php hypertext preprocessor." `http://php.net/`.

[6] "phpmyadmin bringing mysql to the web." `https://www.phpmyadmin.net/`.

[7] "Wampserver, a windows web development environment." `http://www.wampserver.com/en/`.

[8] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.